Subject: Re: Efficiently perform histogram reverse indices like procedure on a string array?
Posted by ben.bighair on Thu, 26 Jul 2012 17:30:37 GMT
View Forum Message <> Reply to Message

On Wednesday, July 25, 2012 10:17:16 PM UTC-4, Jeremy Bailin wrote:
> On 7/25/12 9:09 PM, Bogdanovist wrote:
> &gt; I have an array of a data structure, one tag of which is a string identifier indicating which location the data belongs to. There are many thousands of data points, but only about a dozen or so unique locations.
> &gt;
> &gt; I make frequent use of the HISTOGRAM function with the reverse_indices in order to carve up data by some identifier, most commonly the time. In this case, I want to divide out the data by site efficiently. I can&#39;t use HISTOGRAM on strings, so I need some other approach. There are plenty of ways this can be done, but I&#39;d like some views on the better and most efficient ways to do it.
> &gt;
> &gt; Take an example, say we have a simple string array
> &gt;
> &gt;   foo=[&#39;a&#39;,&#39;b&#39;,&#39;c& #39;,&#39;b&#39;,&#39;b&#39;,&#39;a& #39;,&#39;a&#39;,&#39;c&#39;]
> &gt;
> &gt; To determine the list of unique strings we could do
> &gt;
> &gt;   sfoo = foo[sort(foo)]
> &gt;   print,sfoo[uniq(sfoo)]
> &gt;
> &gt; We can then repeatedly use WHERE to find the indices in the data array(s) corresponding to each site.
> &gt;
> &gt; Is there a quicker/better way to do this? Repeatedly calling WHERE seems inefficient (certainly HISTOGRAM is way faster when it is usable)
>
> Use VALUE_LOCATE to find where in the list of unique indices the
> elements belong to, and use that index as a number that you can run
> HISTOGRAM on.
>
> (raise your hand everyone who saw that coming...)
>
> -Jeremy.

Not me. I had no idea VALUE_LOCATE works on strings. Now that is cool!